Network Management & Monitoring

NAGIOS

These materials are licensed under the Creative Commons *Attribution-Noncommercial 3.0 Unported* license (http://creativecommons.org/licenses/by-nc/3.0/)

Introduction

Network Monitoring Tools

- Availability
- Reliability
- Performance

Nagios actively monitors the availability of devices and services

Introduction

- Possibly the most used open source network monitoring software.
- Has a web interface.
  - Uses CGIs written in C for faster response and scalability.
- Can support up to thousands of devices and services.

Installation

In Debian/Ubuntu

```
# apt-get install nagios3
```

Key directories

```
/etc/nagios3
/etc/nagios3/conf.d
/etc/nagios-plugins/conf
/usr/lib/nagios/plugins
/usr/share/nagios3/htdocs/images/logos
```

Nagios web interface is here:

http://pcN.ws.nsrc.org/nagios3/

Plugins

Plugins are used to verify services and devices:

- Nagios architecture is simple enough that writing new plugins is fairly easy in the language of your choice.
- There are ***many, many*** plugins available (thousands).
  - ✓ http://exchange.nagios.org/
  - ✓ http://nagiosplugins.org/

Features

- Configuration done in text files, based on templates.
- Nagios reads its configuration from a directory. You determine how to divide your configuration files.
- Uses parallel checking and forking for scalability

Features cont.

- Utilizes topology to determine dependencies.
  - Differentiates between what is *down* vs. what is *unreachable*. Avoids running unnecessary checks and sending redundant alarms
- Allows you to define how to send notifications based on combinations of:
  - Contacts and lists of contacts
  - Devices and groups of devices
  - Services and groups of services
  - Defined hours by persons or groups.
  - The state of a service.

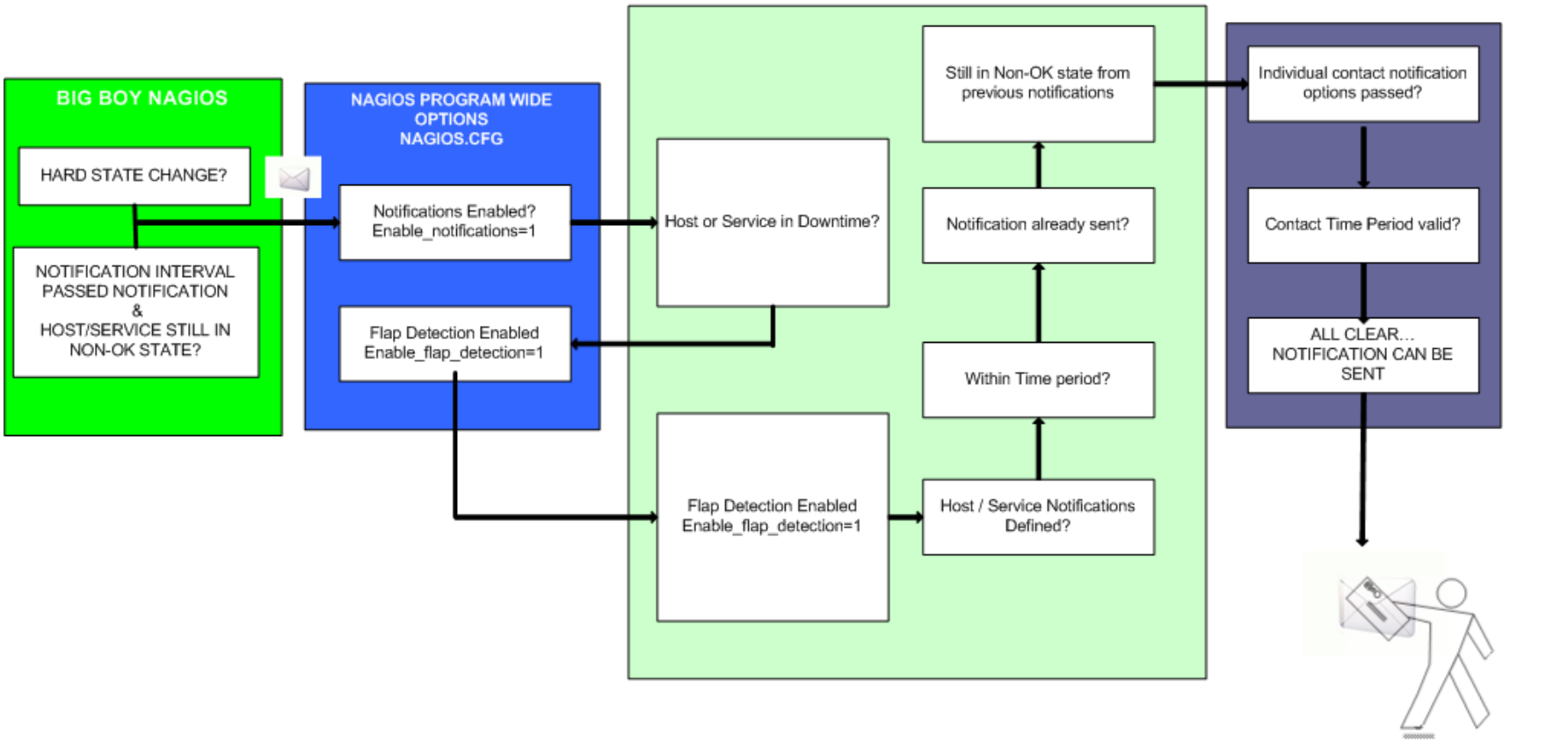
Notification Options (Host)

Host state:

When configuring a host you have the following notification options:

- **d:** DOWN
- **u:** UNREACHABLE
- **r:** RECOVERY
- **f:** FLAPPING
- **n:** NONE

NAGIOS - NOTIFICATION FLOW DIAGRAM



NOTE: The flow will only continue when each of the listed filters are satisfied.

CONTACT GETS THE NOTIFICATION MESSAGE

How checks work

- A node/host/device consists of one or more service checks (PING, HTTP, MYSQL, SSH, etc.)
- Periodically Nagios checks each service for each node and determines if state has changed. State changes are:
  - CRITICAL
  - WARNING
  - UNKNOWN
- For each state change you can assign:
  - Notification options (as mentioned before)
  - Event handlers

How checks work continued

Parameters

- Normal checking interval
- Re-check interval
- Maximum number of checks.
- Period for each check

- Node checks only happen when services respond.
  - A node can be:
    - DOWN
    - UNREACHABLE

How checks work continued

- By default Nagios does a node check 3 times before it will change the node's state to down.
- No response states goes to *warning* then *critical*

The concept of “parents”

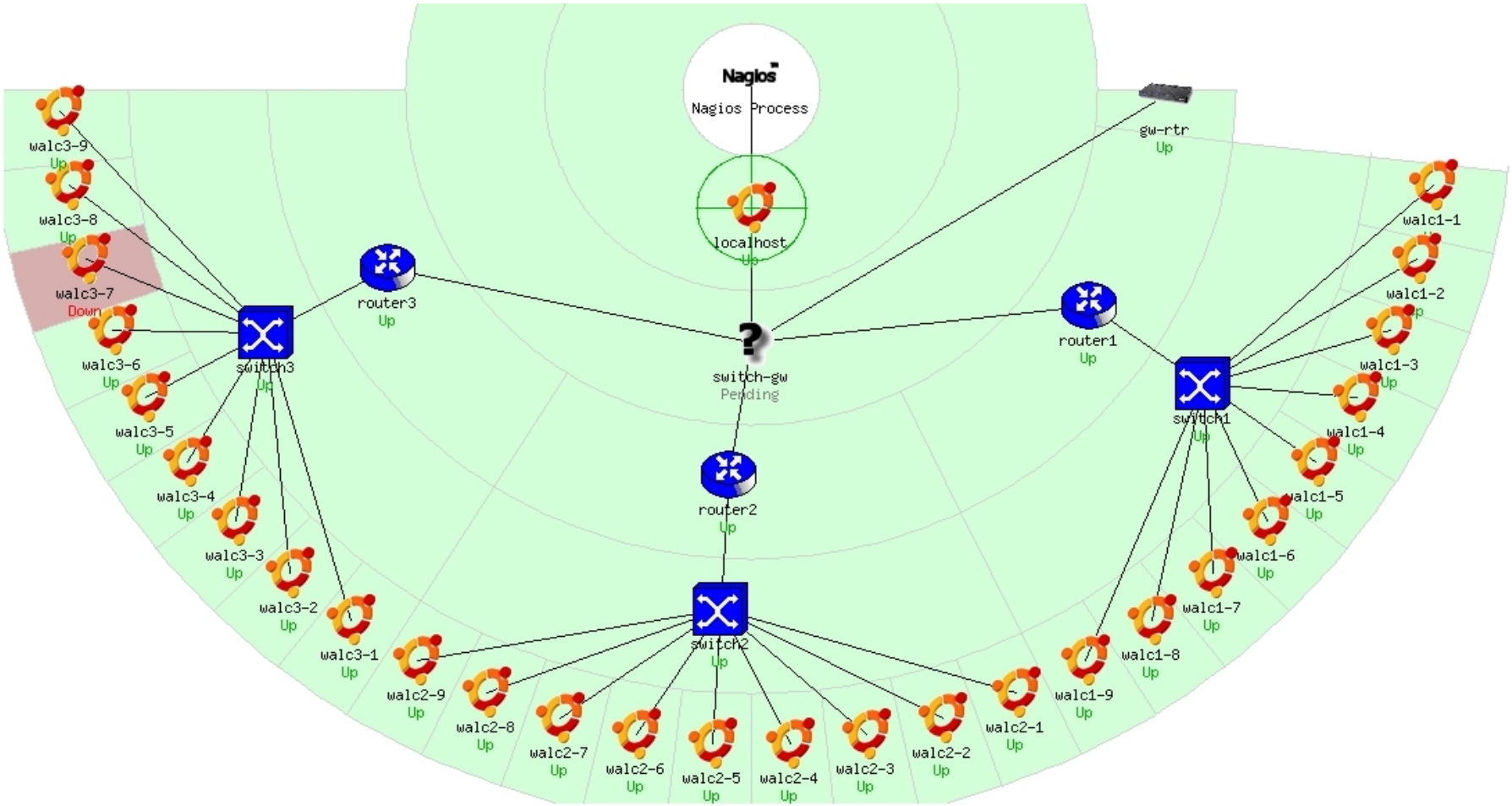
Nodes can have parents:

- The parent of a PC connected to a switch would be the switch.
- Allows us to specify the dependencies between devices.
- Avoids sending alarms when parent does not respond.
- A node can have multiple parents (dual homed).

Network viewpoint

- Where you locate your Nagios server will determine your point of view of the network.
- The Nagios server becomes the “root” of your dependency tree

Network viewpoint

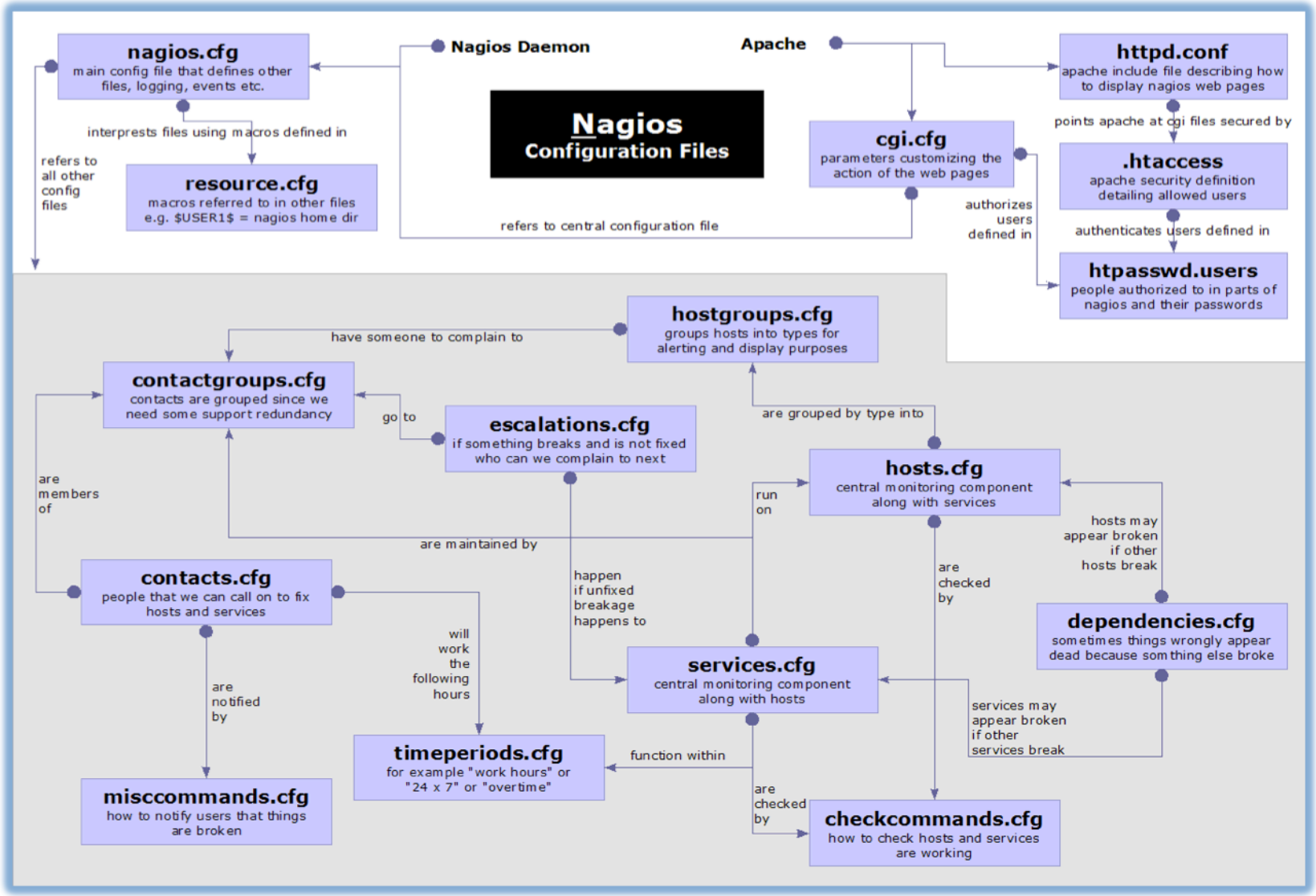


Demo Nagios

Configuration Files

- Lots!
- Can seem complex at first
- **Object oriented**
  - Objects (devices or services) inherit attributes.
  - Apply functionality to *groups of devices.*
  - Do not apply functionality to individual objects. Does not scale!
  - Once you understand Nagios configs the rest is easy…

Configuration files (Official)



Configuration Files

Located in /etc/nagios3/

Important files include:

- cgi.cfg — Controls the web interface and security options.
- commands.cfg — The commands that Nagios uses for notifications.
- nagios.cfg — Main configuration file.
- conf.d/* — All other configuration goes here!

Configuration files continued

Under conf.d/*

contacts_nagios2.cfg	users and groups
extinfo_nagios2.cfg	make your UI pretty
generic-host_nagios2.cfg	default host template
generic-service_nagios2.cfg	default service template
host-gateway_nagios3.cfg	host at default gw definition
hostgroups_nagios2.cfg	groups of nodes
localhost_nagios2.cfg	definition of nagios host
services_nagios2.cfg	what services to check
timeperiods_nagios2.cfg	when to check who to notify

Configuration files continued

Under conf.d some other possible config files:

- servicegroups.cfg — Groups of nodes and services
- pcs.cfg — Sample definition of PCs (hosts)
- switches.cfg — Definitions of switches (hosts)
- routers.cfg — Definitions of routers (hosts)

Pre-installed plugins in Ubuntu

/usr/lib/nagios/plugins

check_apt	check_file_age	check_jabber	check_nntp	check_procs	check_swap
check_bgpstate	check_flexlm	check_ldap	check_nntps	check_radius	check_tcp
check_breeze	check_ftp	check_ldaps	check_nt	check_real	check_time
check_by_ssh	check_host	check_linux_raid	check_ntp	check_rpc	check_udp
check_clamd	check_hpjd	check_load	check_ntp_peer	check_rta_multi	check_ups
check_cluster	check_http	check_log	check_ntp_time	check_sensors	check_users
check_dhcp	check_icmp	check_mailq	check_nwstat	check_simap	check_wave
check_dig	check_ide_smart	check_mrtg	check_oracle	check_smtp	negate
check_disk	check_ifoperstatus	check_mrtgtraf	check_overcr	check_snmp	urlize
check_disk_smb	check_ifstatus	check_mysql	check_pgsql	check_spop	utils.pm
check_dns	check_imap	check_mysql_query	check_ping	check_ssh	utils.sh
check_dummy	check_ircd	check_nagios	check_pop	check_ssmtp	

/etc/nagios-plugins/config

apt.cfg	disk-smb.cfg	ftp.cfg	ldap.cfg	mysql.cfg	ntp.cfg	radius.cfg	ssh.cfg
breeze.cfg	dns.cfg	hppjd.cfg	load.cfg	netware.cfg	pgsql.cfg	real.cfg	tcp_udp.cfg
dhcp.cfg	dummy.cfg	http.cfg	mail.cfg	news.cfg	ping.cfg	rpc-nfs.cfg	telnet.cfg
disk.cfg	flexlm.cfg	ifstatus.cfg	mrtg.cfg	nt.cfg	procs.cfg	snmp.cfg	users.cfg

Nodes and services configuration

Based on templates

- This saves lots of time avoiding repetition
- Similar to Object Oriented programming

Create default templates with default parameters for a:

- *generic node* (generic-host_nagios2.cfg)
- *generic service* (generic-service_nagios2.cfg)
- generic contact (contacts_nagios2.cfg)

Generic node template

generic-host_nagios2.cfg

```
define host{
        name                            generic-host    ; The name of this host template
        notifications_enabled           1               ; Host notifications are enabled
        event_handler_enabled           1               ; Host event handler is enabled
        flap_detection_enabled          1               ; Flap detection is enabled
        failure_prediction_enabled      1               ; Failure prediction is enabled
        process_perf_data               1               ; Process performance data
        retain_status_information       1               ; Retain status information across program restarts
        retain_nonstatus_information    1               ; Retain non-status information across program restarts
                check_command                   check-host-alive
                max_check_attempts              10
                notification_interval           0
                notification_period             24x7
                notification_options            d,u,r
                contact_groups                  admins
        register                        0       ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL HOST, JUST A TEMPLATE!
        }
```

Individual node configuration

```
define host{
        use                     generic-host
        host_name               gw-rtr
        alias                   Main workshop router
        address                 192.0.2.1
        contact_groups          router_group
}
```

Generic service configuration

generic-service_nagios2.cfg

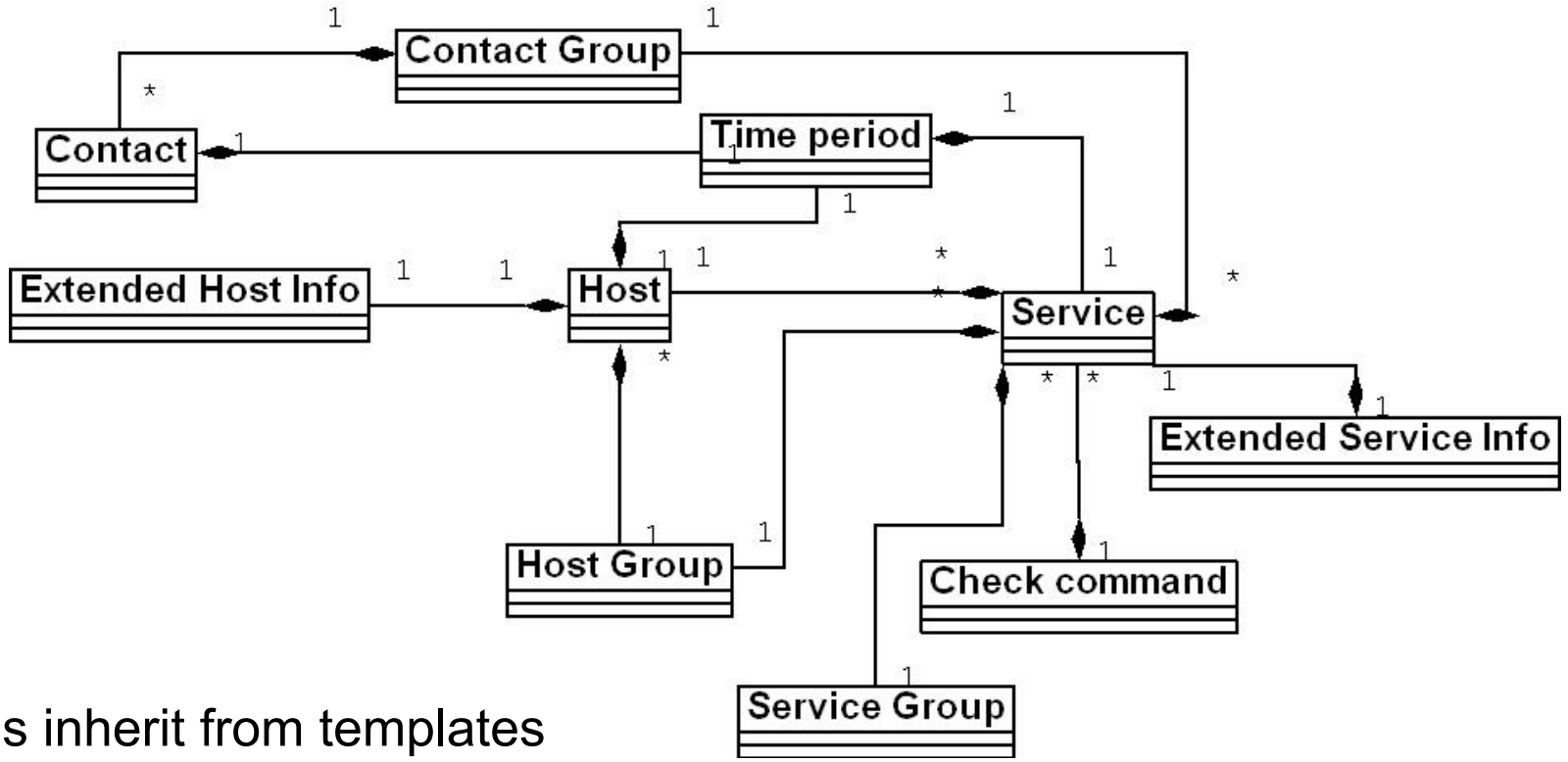
```
define service{
        name                            generic-service
        active_checks_enabled           1
        passive_checks_enabled          1
        parallelize_check               1
        obsess_over_service             0
        check_freshness                 1
        notifications_enabled           1
        event_handler_enabled           1
        flap_detection_enabled          1
        process_perf_data               1
        retain_status_information       1
        retain_nonstatus_information    1
        is_volatile                     0
        check_period                    24x7
        max_check_attempts              5
        normal_check_interval           5
        retry_check_interval            1
        notification_interval           60
        notification_period             24x7
        notification_options            c,r
        register                        0
        }
```

Individual service configuration

```
define service{
        hostgroup_name                  servers
        service_description             PING
        check_command                   check-host-alive
        use                             generic-service
        max_check_attempts              5
        normal_check_interval           5
        notification_options            c,r,f
        notification_interval           0 ; set > 0 if you want to be renotified
}
```

c: Critical
r: Recovering
f: Flapping

Configuration flow

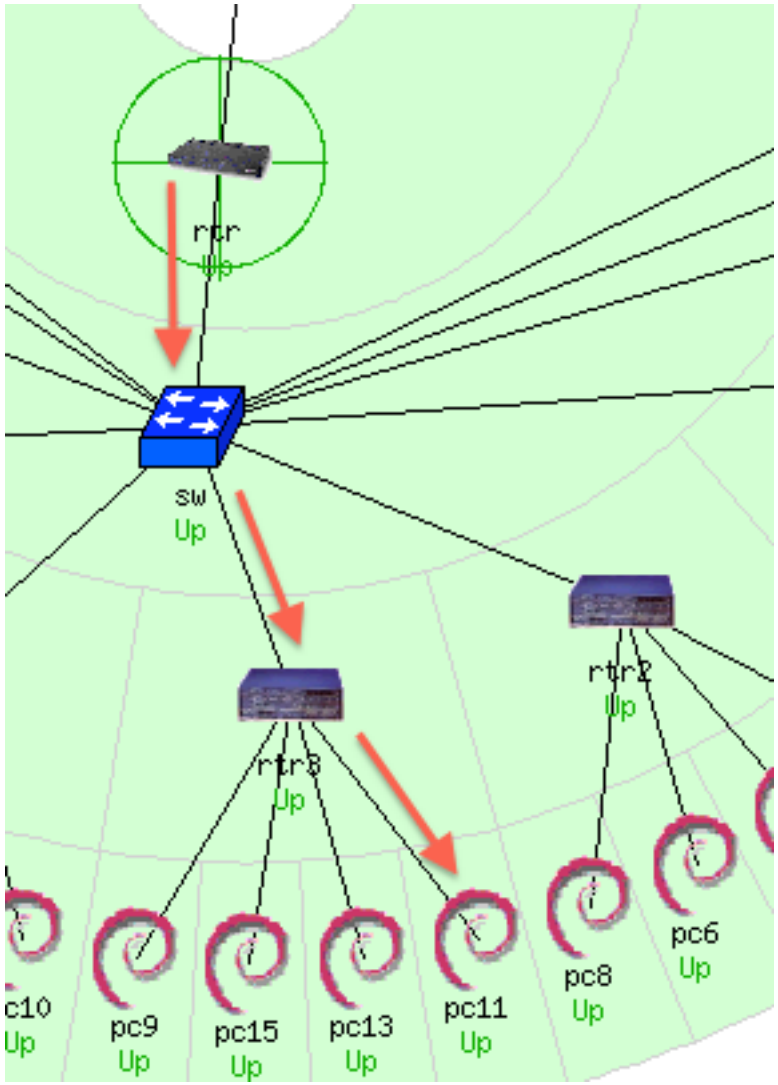


Items inherit from templates

We start with a host

- Place multiple hosts in a group
- Define parents
- Add a service check to the group
- Add extended info, if any

Another view of configuration



RTR

```
define host {
    use             generic-host
    host_name       rtr
    alias           Gateway Router
    address         10.10.0.254     }
```

SW

```
define host {
    use             generic-host
    host_name       sw
    alias           Backbone Switch
    address         10.10.0.253
    parents         rtr     }
```

RTR3

```
define host {
    use             generic-host
    host_name       rtr3
    alias           router 3
    address         10.10.3.254
    parents         sw      }
```

PC11…

OoB notifications

A critical item to remember: an SMS or message system that is independent from your network.

- You can utilize a cell phone connected to the Nagios server
- You can use packages like:

  gnokii: http://www.gnokii.org/
  qpage: http://www.qpage.org/
  sendpage: http://www.sendpage.org/

References

- **Nagios web site**
  http://www.nagios.org/
- **Nagios plugins site**
  http://www.nagiosplugins.org/
- *Nagios System and Network Monitoring*, by Wolfgang Barth. Good book about Nagios.
- **Unofficial Nagios plugin site**
  http://nagios.exchange.org/
- **A Debian tutorial on Nagios**
  http://www.debianhelp.co.uk/nagios.htm
- **Commercial Nagios support**
  http://www.nagios.com/

Questions?

?

Additional Details

A few additional slides you may find useful or informative…

Features, features, features…

- Allows you to acknowledge an event.
  - A user can add comments via the GUI
- You can define maintenance periods
  - By device or a group of devices
- Maintains availability statistics.
- Can detect flapping and suppress additional notifications.
- Allows for multiple notification methods:
  - e-mail, pager, SMS, winpopup, audio, etc...
- Allows you to define notification levels for escalation

Main configuration details

Global settings

File: `/etc/nagios3/nagios.cfg`

- Says where other configuration files are.
- General Nagios behavior:
  - For large installations you should tune the installation via this file.
  - See: *Tunning Nagios for Maximum Performance* **http://nagios.sourceforge.net/docs/2_0/tuning.html**

CGI configuration

/etc/nagios3/cgi.cfg

- You can change the CGI directory if you wish
- Authentication and authorization for Nagios use:
  - Activate authentication via Apache's .htpasswd mechanism, or using RADIUS or LDAP.
  - Users can be assigned rights via the following variables:
    - authorized_for_system_information
    - authorized_for_configuration_information
    - authorized_for_system_commands
    - authorized_for_all_services
    - authorized_for_all_hosts
    - authorized_for_all_service_commands
    - authorized_for_all_host_commands

Time Periods

This defines the base periods that control checks, notifications, etc.

- Defaults: 24 x 7
- Could adjust as needed, such as work-week only.
- Could adjust a new time period for "outside of regular hours", etc.

```
# '24x7'
define timeperiod{
        timeperiod_name 24x7
        alias           24 Hours A Day, 7 Days A Week
        sunday          00:00-24:00
        monday          00:00-24:00
        tuesday         00:00-24:00
        wednesday       00:00-24:00
        thursday        00:00-24:00
        friday          00:00-24:00
        saturday        00:00-24:00
        }
```

Configuring service/host checks:

Definition of “host alive”

```
# 'check-host-alive' command definition
define command{
        command_name    check-host-alive
        command_line    $USER1$/check_ping -H $HOSTADDRESS$ -w 2000.0,60% -c
5000.0,100% -p 1 -t 5
 }
```

- Located in /etc/nagios-plugins/config, then adjust in /etc/nagios3/conf.d/ services_nagios2.cfg
- While these are “service” or “host” checks Nagios refers to them as “commands”

Notification commands

Allows you to utilize any command you wish.
We could use this to generate tickets in RT.

```
# 'notify-by-email' command definition
define command{
        command_name    notify-by-email
        command_line    /usr/bin/printf "%b" "Service: $SERVICEDESC$\nHost:
$HOSTNAME$\nIn: $HOSTALIAS$\nAddress: $HOSTADDRESS$\nState: $SERVICESTATE$
\nInfo: $SERVICEOUTPUT$\nDate: $SHORTDATETIME$" | /bin/mail -s
'$NOTIFICATIONTYPE$: $HOSTNAME$/$SERVICEDESC$ is $SERVICESTATE$'
$CONTACTEMAIL$
        }
```

From: nagios@nms.localdomain
To: router_group@localdomain
Subject: Host DOWN alert for TLD1-RTR!
Date: Thu, 29 Jun 2006 15:13:30 -0700

Host: gw-rtr
In: Core_Routers
State: DOWN
Address: 192.0.2.100
Date/Time: 06-29-2006 15:13:30
Info: CRITICAL - Plugin timed out after 6 seconds

Group service configuration

```
# check that ssh services are running
define service {
        hostgroup_name          ssh-servers
        service_description     SSH
        check_command           check_ssh
        use                     generic-service
        notification_interval   0 ; set > 0 if you want to be renotified
}
```

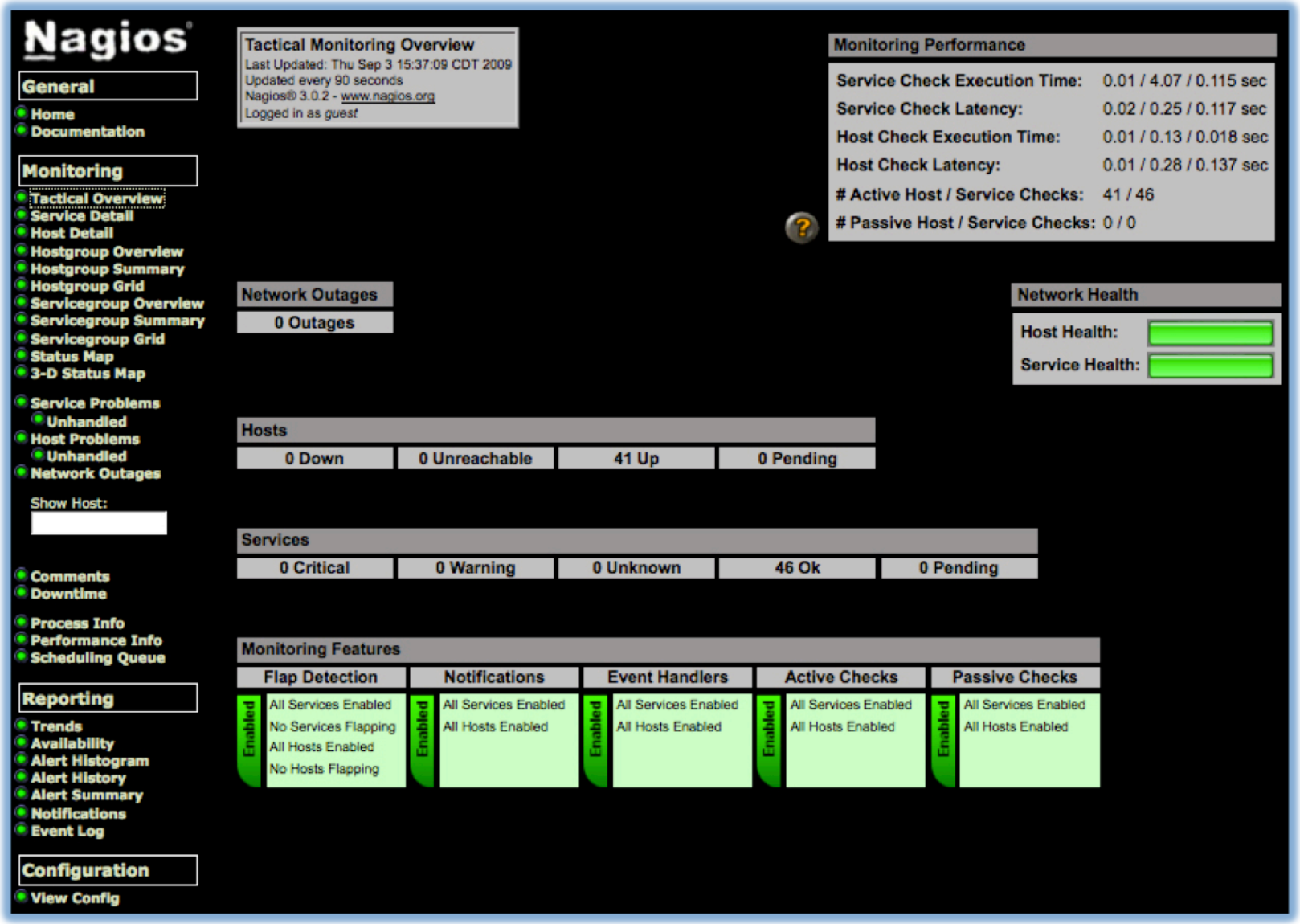
The “service_description” is important if you plan to create Service Groups. Here is a sample Service Group definition:

```
define servicegroup{
        servicegroup_name       Webmail
        alias                   web-mta-storage-auth
        members                 srvr1,HTTP,srvr1,SMTP,srvr1,POP,srvr1,IMAP,
                                 srvr1,RAID,srvr1,LDAP, srvr2,HTTP,srvr2,SMTP,
                                srvr2,POP,srvr2,IMAP,srvr2,RAID,srvr2,LDAP
}
```

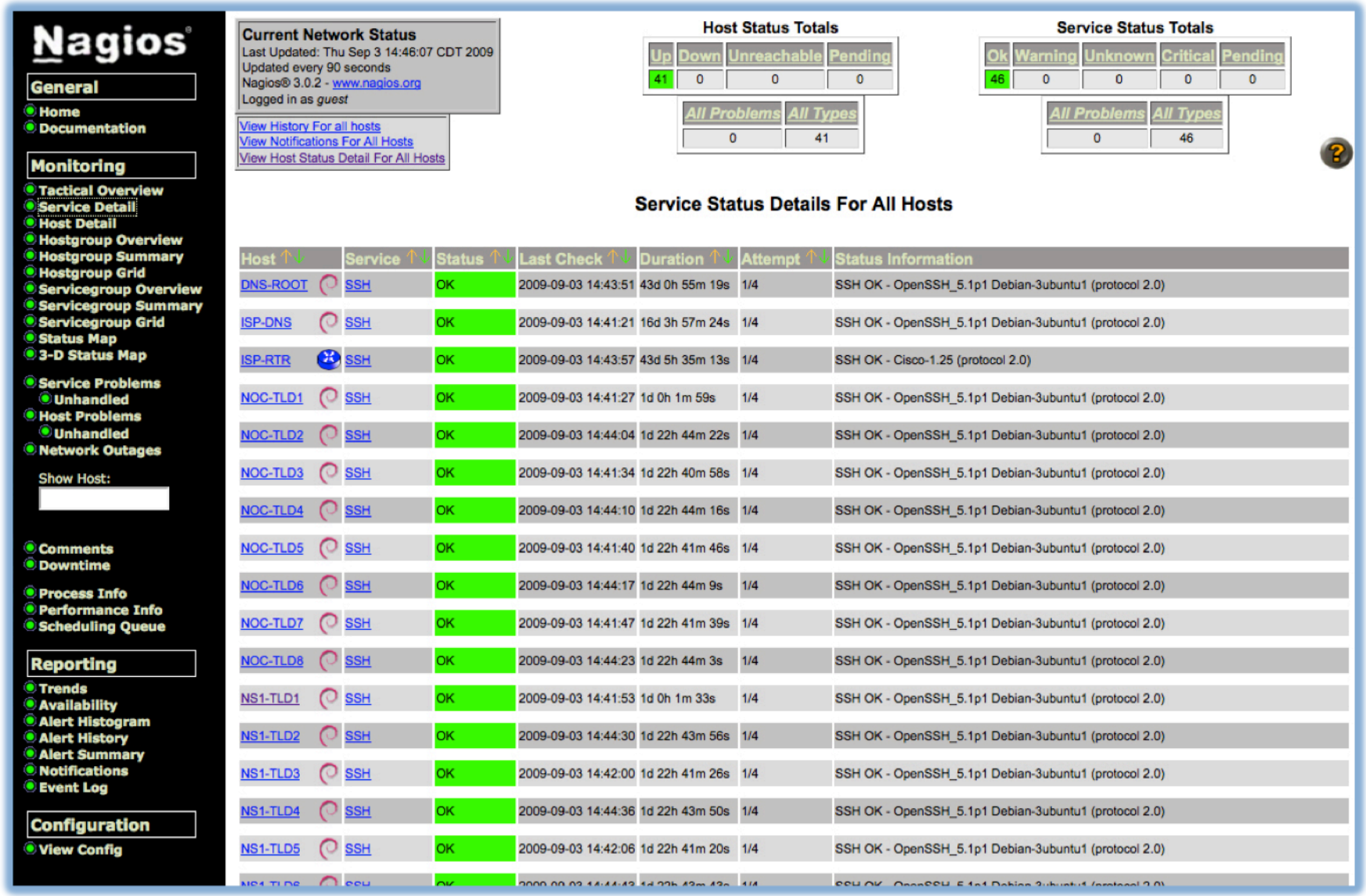
Screen Shots

A few sample screen shots from a Nagios install.

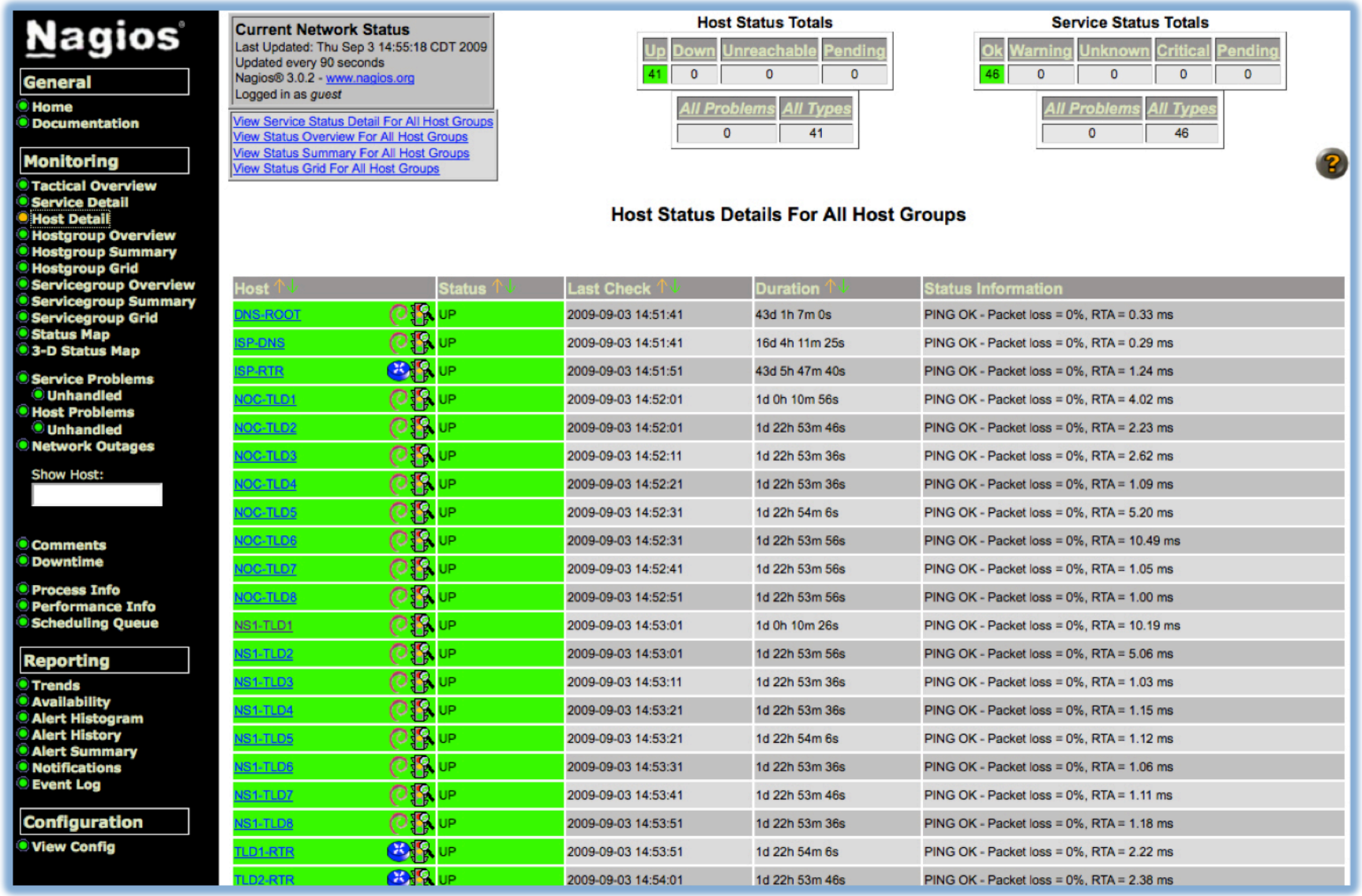
General View



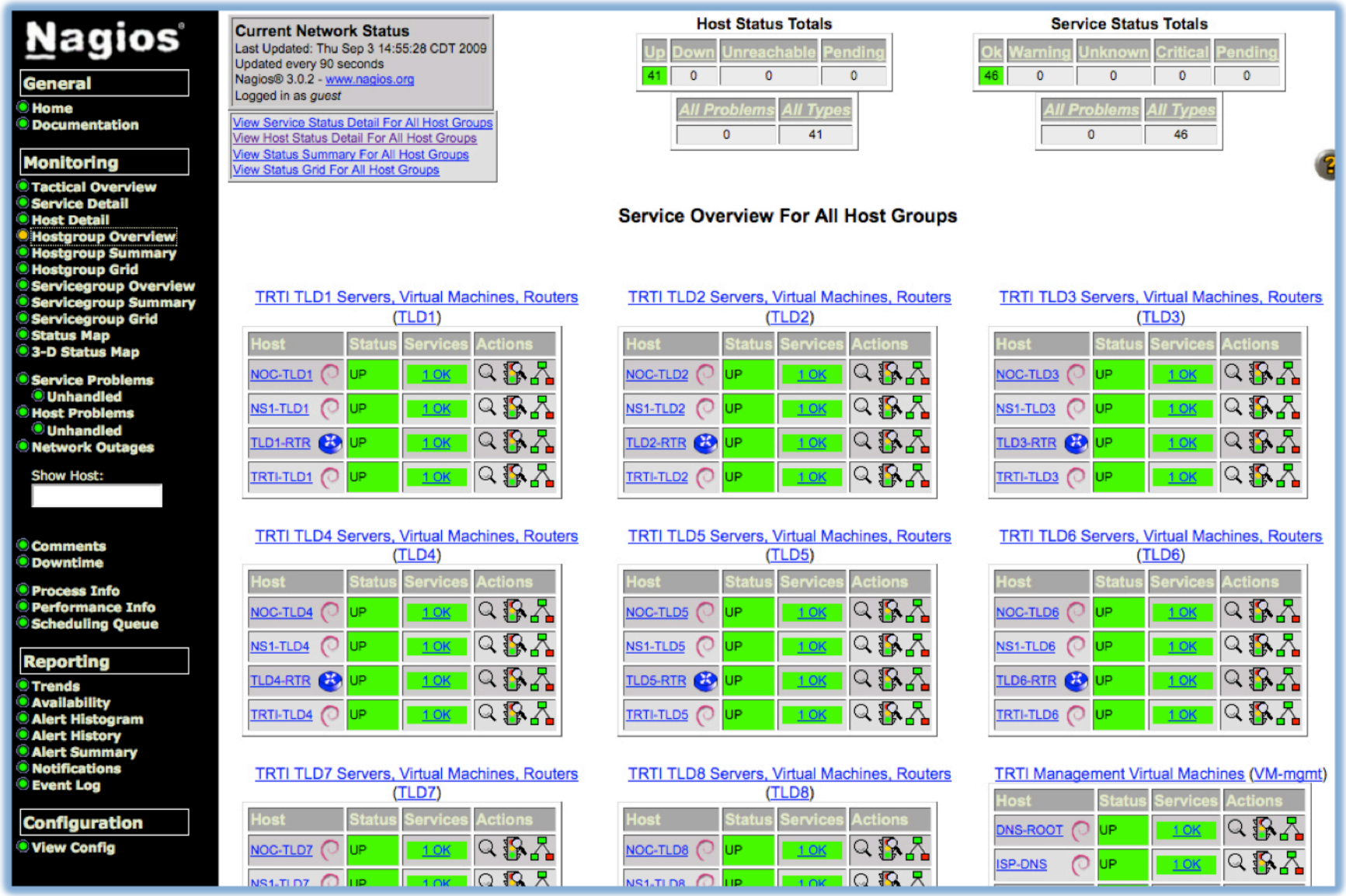
Service Detail



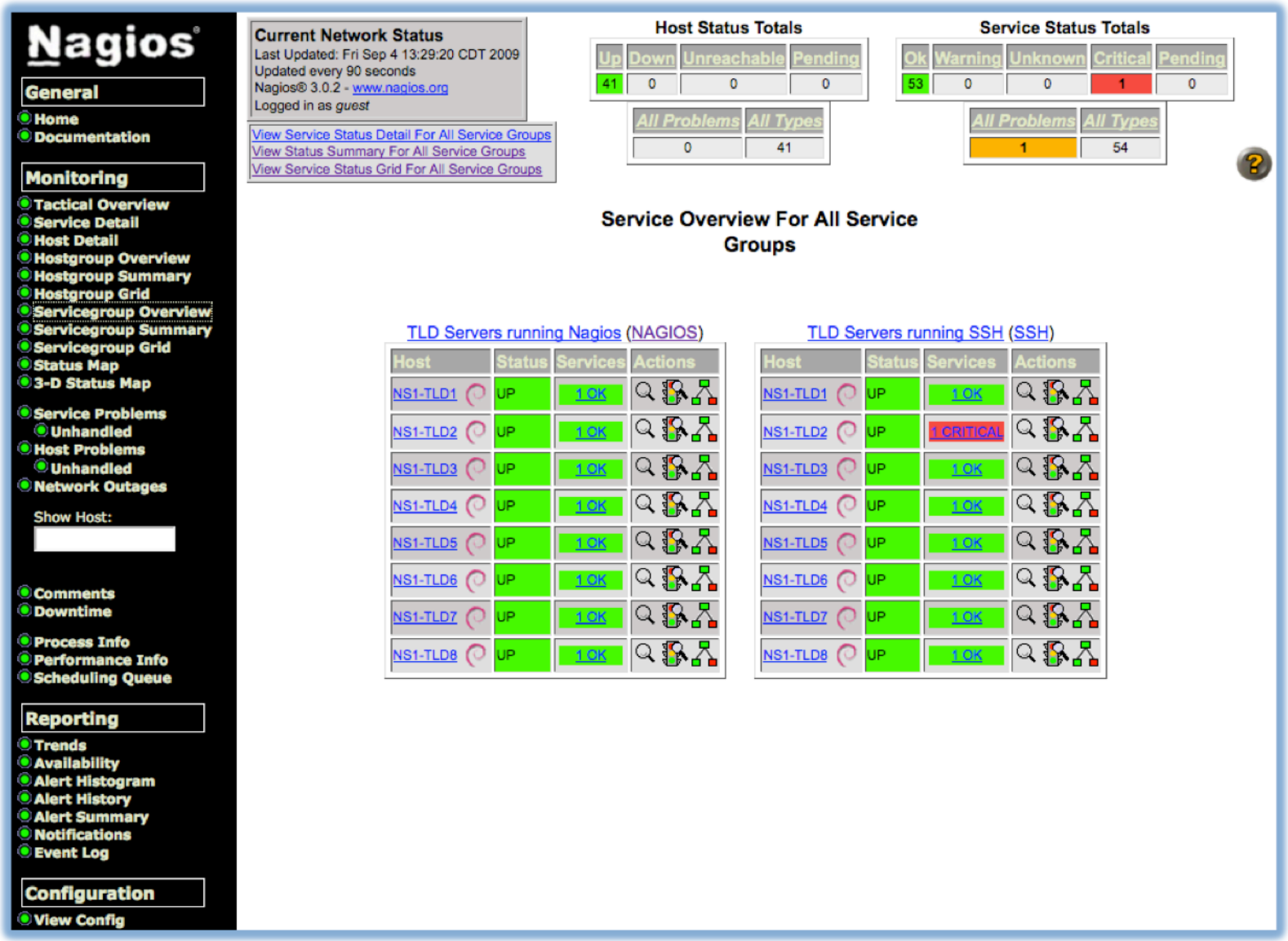
Host Detail


Nagios
General
Home
Documentation
Monitoring
Tactical Overview
Service Detail
Host Detail
Hostgroup Overview
Hostgroup Summary
Hostgroup Grid
Servicegroup Overview
Servicegroup Summary
Servicegroup Grid
Status Map
3-D Status Map
Service Problems
Unhandled
Host Problems
Unhandled
Network Outages
Show Host:
Comments
Downtime
Process Info
Performance Info
Scheduling Queue
Reporting
Trends
Availability
Alert Histogram
Alert History
Alert Summary
Notifications
Event Log
Configuration
View Config
Current Network Status
Last Updated: Thu Sep 3 14:55:18 CDT 2009
Updated every 90 seconds
Nagios® 3.0.2 - www.nagios.org
Logged in as guest
View Service Status Detail For All Host Groups
View Status Overview For All Host Groups
View Status Summary For All Host Groups
View Status Grid For All Host Groups
Host Status Totals
Up Down Unreachable Pending
41 0 0 0
All Problems All Types
0 41
Service Status Totals
Ok Warning Unknown Critical Pending
46 0 0 0 0
All Problems All Types
0 46
Host Status Details For All Host Groups
Host Status Last Check Duration Status Information
DNS-ROOT UP 2009-09-03 14:51:41 43d 1h 7m 0s PING OK - Packet loss = 0%, RTA = 0.33 ms
ISP-DNS UP 2009-09-03 14:51:41 16d 4h 11m 25s PING OK - Packet loss = 0%, RTA = 0.29 ms
ISP-RTR UP 2009-09-03 14:51:51 43d 5h 47m 40s PING OK - Packet loss = 0%, RTA = 1.24 ms
NOC-TLD1 UP 2009-09-03 14:52:01 1d 0h 10m 56s PING OK - Packet loss = 0%, RTA = 4.02 ms
NOC-TLD2 UP 2009-09-03 14:52:01 1d 22h 53m 46s PING OK - Packet loss = 0%, RTA = 2.23 ms
NOC-TLD3 UP 2009-09-03 14:52:11 1d 22h 53m 36s PING OK - Packet loss = 0%, RTA = 2.62 ms
NOC-TLD4 UP 2009-09-03 14:52:21 1d 22h 53m 36s PING OK - Packet loss = 0%, RTA = 1.09 ms
NOC-TLD5 UP 2009-09-03 14:52:31 1d 22h 54m 6s PING OK - Packet loss = 0%, RTA = 5.20 ms
NOC-TLD6 UP 2009-09-03 14:52:31 1d 22h 53m 56s PING OK - Packet loss = 0%, RTA = 10.49 ms
NOC-TLD7 UP 2009-09-03 14:52:41 1d 22h 53m 56s PING OK - Packet loss = 0%, RTA = 1.05 ms
NOC-TLD8 UP 2009-09-03 14:52:51 1d 22h 53m 56s PING OK - Packet loss = 0%, RTA = 1.00 ms
NS1-TLD1 UP 2009-09-03 14:53:01 1d 0h 10m 26s PING OK - Packet loss = 0%, RTA = 10.19 ms
NS1-TLD2 UP 2009-09-03 14:53:01 1d 22h 53m 56s PING OK - Packet loss = 0%, RTA = 5.06 ms
NS1-TLD3 UP 2009-09-03 14:53:11 1d 22h 53m 36s PING OK - Packet loss = 0%, RTA = 1.03 ms
NS1-TLD4 UP 2009-09-03 14:53:21 1d 22h 53m 36s PING OK - Packet loss = 0%, RTA = 1.15 ms
NS1-TLD5 UP 2009-09-03 14:53:21 1d 22h 54m 6s PING OK - Packet loss = 0%, RTA = 1.12 ms
NS1-TLD6 UP 2009-09-03 14:53:31 1d 22h 53m 36s PING OK - Packet loss = 0%, RTA = 1.06 ms
NS1-TLD7 UP 2009-09-03 14:53:41 1d 22h 53m 46s PING OK - Packet loss = 0%, RTA = 1.11 ms
NS1-TLD8 UP 2009-09-03 14:53:51 1d 22h 53m 36s PING OK - Packet loss = 0%, RTA = 1.18 ms
TLD1-RTR UP 2009-09-03 14:53:51 1d 22h 54m 6s PING OK - Packet loss = 0%, RTA = 2.22 ms
TLD2-RTR UP 2009-09-03 14:54:01 1d 22h 53m 46s PING OK - Packet loss = 0%, RTA = 2.38 ms

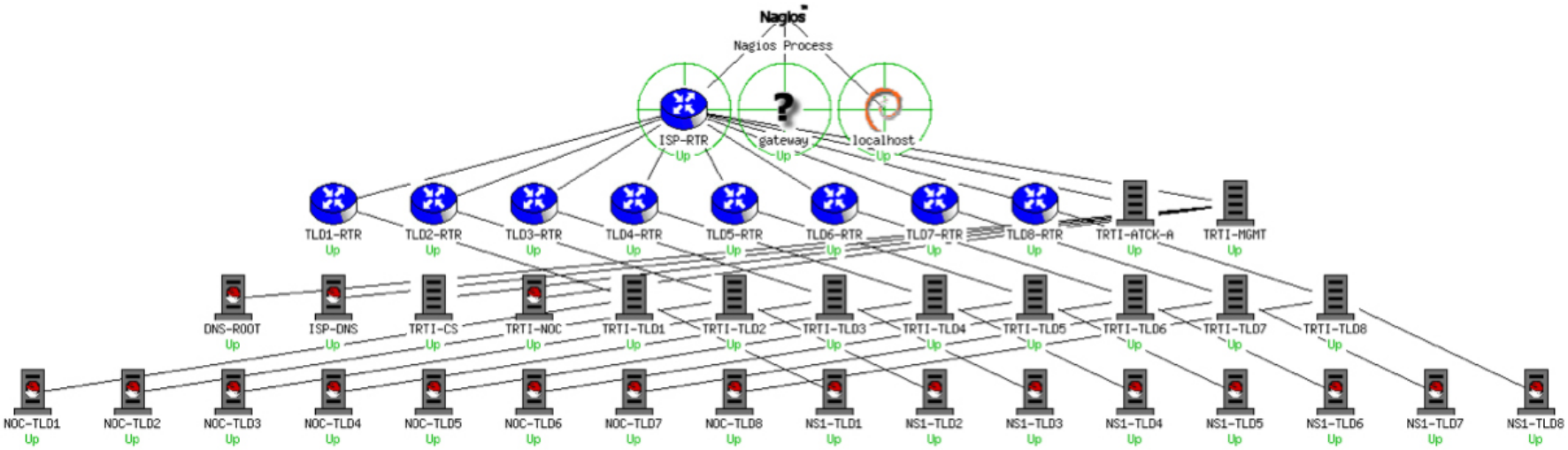
Host Groups Overview



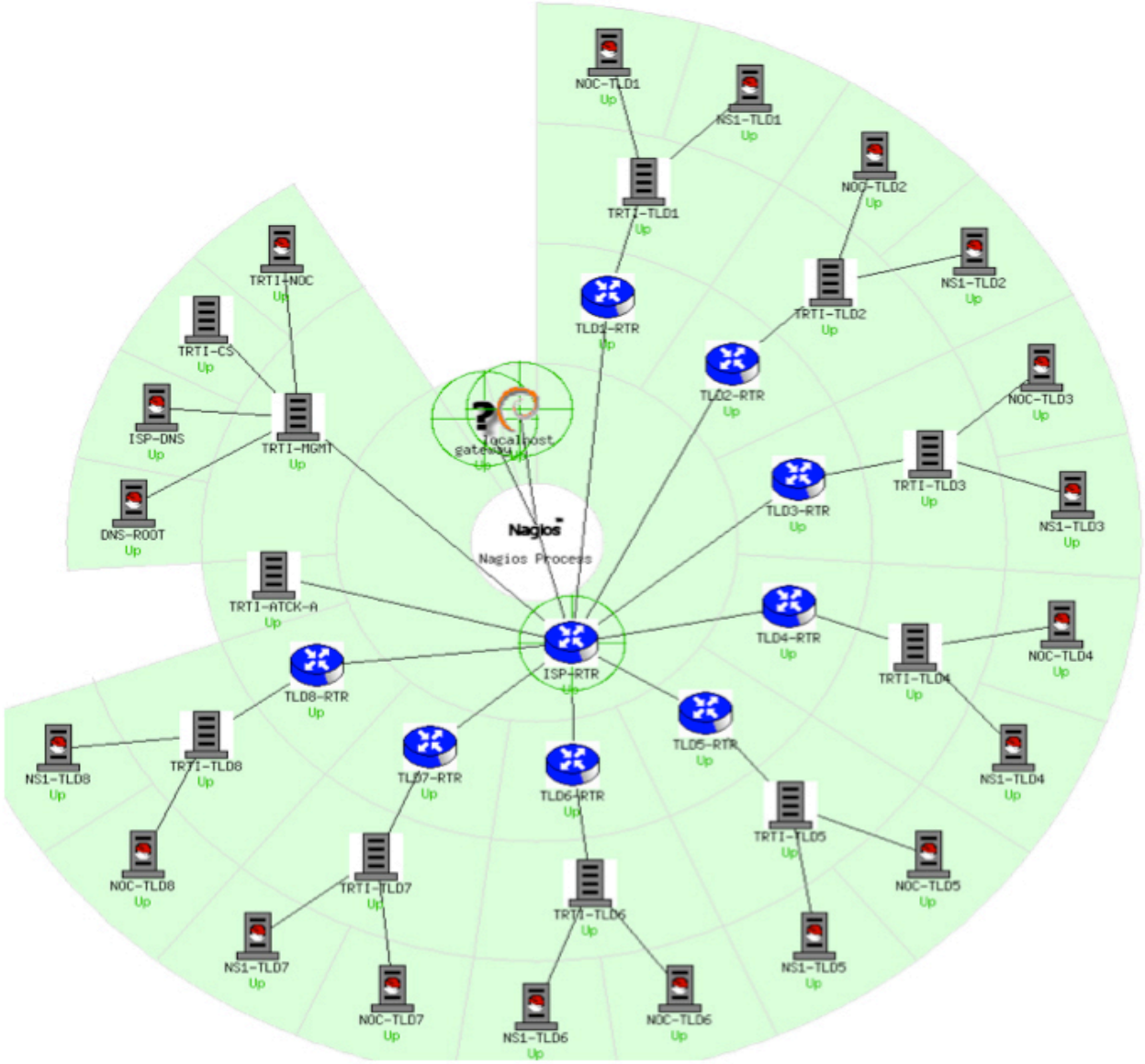
Service Groups Overview



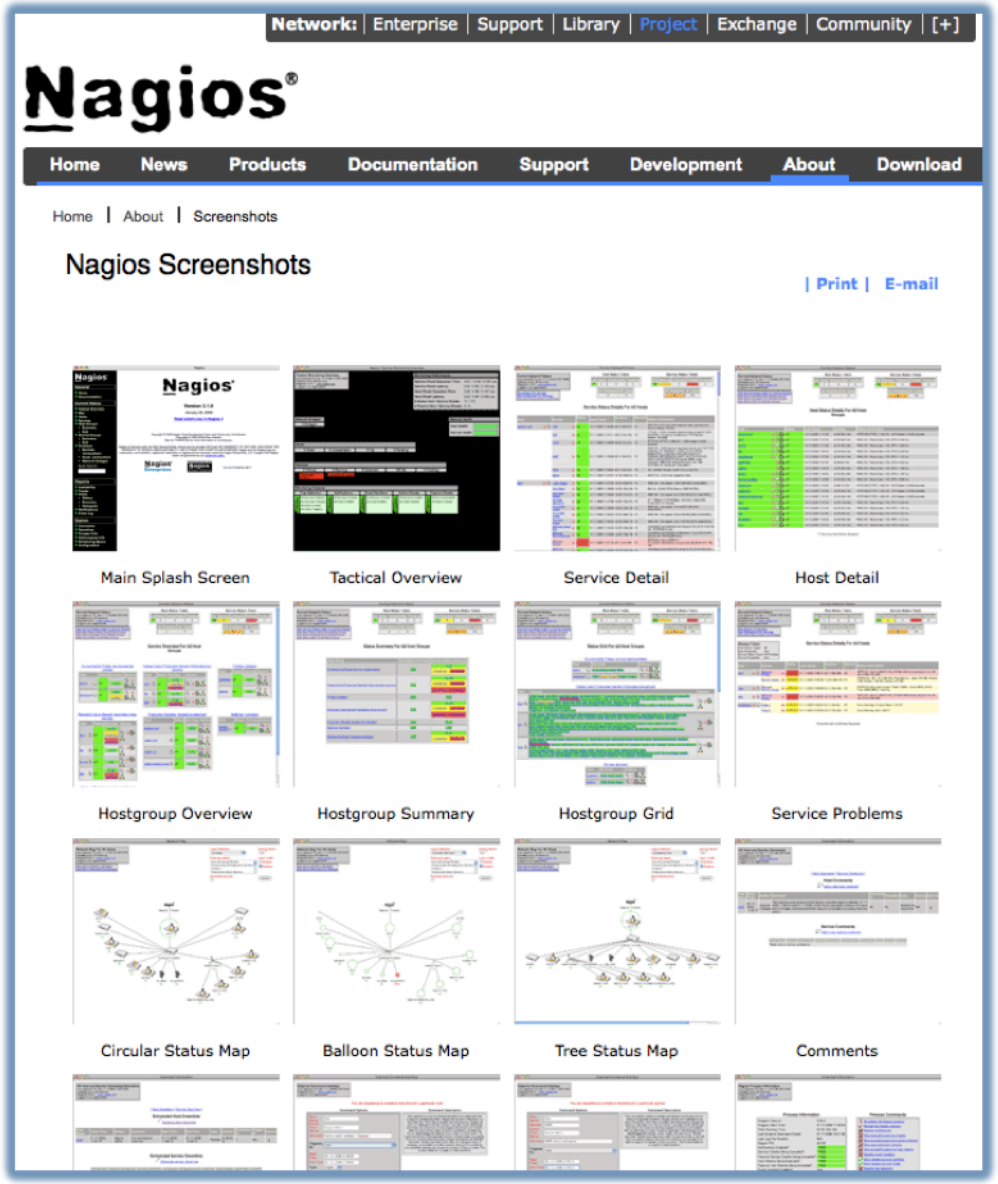
Collapsed tree status map



Marked-up circular status map



More sample screenshots



Many more sample Nagios screenshots available here:

http://www.nagios.org/about/screenshots